3rd Sem Mini Project Report on

Speech-Text-emotion Recognition on voice messages

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Student Name	University Roll No.
Anmol Rawat	2023251

Under the Guidance of
Dr.Pramod Mehra
Assistant Professor

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand

2024-25

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Speech-Text-Emotion Recognition on Voice Messages"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Dr. Pramod Mehra, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Anmol Rawat University Roll no :2023251

The above mentioned student shall be working under the supervision of the undersigned on the **"Speech-Text-Emotion-Recognition on Voice Messages"**

Supervisor **Head of the Department**

Examination

Name of the Examiners: **Signature with Date**

1.

2.

Table of Content

Chapter 1

Introduction and Problem Statement

1.1 Introduction:

Emotion recognition through voice messages has rapidly emerged as a critical area in human-computer interaction (HCI) and affective computing. Emotions play a fundamental role in human communication, influencing how people convey meaning and interpret messages. Traditionally, emotional expression has been studied in the context of facial expressions, body language, and written text. However, speech, as a natural form of human interaction, provides a wealth of cues about a person's emotional state, such as tone, pitch, cadence, and rhythm. Recognizing emotions through speech can vastly enhance the quality of interactions in numerous domains, from virtual assistants and customer service systems to mental health support tools and personalized feedback mechanisms.

In an increasingly interconnected world, where voice communication through smartphones, social media platforms, and virtual assistants has become ubiquitous, emotion recognition in voice messages holds immense potential. For example, detecting emotional distress in a customer service call or recognizing a user's emotional state in a messaging platform can enable more effective, empathetic, and context-aware responses. Despite its promise, emotion recognition remains a challenging task due to the inherent complexity of speech patterns, individual variations in voice, accent, language, and cultural context.

This project aims to develop a robust emotion recognition system that analyzes voice messages to identify emotions. The system will employ advanced speech-to-text conversion techniques and powerful emotion classification models to interpret emotional cues from speech. By utilizing state-of-the-art machine learning methodologies, the system will first transcribe audio inputs into text and then classify the underlying emotion expressed within the message. This technology has the potential to revolutionize communication systems by providing deeper insights into

user emotions, ultimately enhancing interactions across a variety of platforms.

1.2 Problem Statement:

In today's digital communication era, platforms like **Discord** enable users to connect through voice and text messages. However, the lack of emotional context in textual communication often leads to misunderstandings, while real-time voice-based emotion recognition remains an underutilized tool for enhancing human connection in online interactions.

This project aims to develop a **Speech-to-Text Emotion Recognition Model** integrated with a Discord bot to analyze voice messages, transcribe them into text, and accurately detect the underlying emotional tone. By incorporating real-time emotion analysis, the system will provide users with an enhanced communication experience by appending emotion-specific emojis or descriptive tags to messages.

By bridging the gap between voice intonation and digital communication, this project should enhance the richness of online interactions, enabling users to better understand and respond to the emotions of their peers in real-time.

Chapter 2

Methodology

This section outlines the step-by-step process followed in building and training the emotion recognition system, which is designed to analyze emotions in both text and voice messages. The methodology integrates machine learning techniques with a multilingual approach to ensure that the system works effectively across both English and Hindi languages. Each step, from data collection to model evaluation, was carefully planned to ensure robust performance in real-world applications.

2.1 Dataset Collection:

For the emotion recognition task, we used the **GoEmotion dataset [2]**, which includes more than **60,000 text samples** annotated with **27+ emotions**. However, we considered only the **basic emotions**: **happiness, sadness, anger, fear, surprise, disgust, and neutral**.

This selection was made to **reduce model complexity**, address potential **class imbalance issues**, and focus on the most **universally recognized emotional categories** for improved performance and interpretability.

This dataset provides a **rich source of text-based emotion labels**, which are essential for training a **supervised machine learning model**. We preprocessed the dataset to ensure **clean and balanced data** for optimal training performance. Text data was vectorized using **TF-IDF (Term Frequency-Inverse Document Frequency)**, and class imbalance was addressed using **SMOTE (Synthetic Minority Over-sampling Technique)** to ensure fair representation across all emotion categories.

By carefully curating and preprocessing the GoEmotion dataset, we were able to build a reliable emotion recognition model capable of accurately predicting emotions in both **text and voice inputs**.

2.2 Data Preprocessing Data:

Preprocessing plays a crucial role in ensuring that the input data is in the right format for the machine learning model. The preprocessing steps included the following:

- **Speech-to-Text Conversion:** For audio data, we employed the SpeechRecognition library to convert voice messages into text. This step enabled us to analyze emotions from spoken inputs as text.
- **Text Cleaning and Tokenization:** The transcribed text underwent cleaning, where unnecessary punctuation, stop words, and irrelevant symbols were removed. After cleaning, the text was tokenized into smaller units (words or phrases) for further processing.
- **Multilingual Support with Deep-translator:** To handle multilingual text inputs, particularly for English and Hindi, Deep-translator facilitated easy management of language-specific data.
- **Feature Extraction:** For text data, TF-IDF (Term Frequency-Inverse Document Frequency) was used as a feature extraction technique to capture important features from the text.
- **Data Augmentation:** Techniques such as paraphrasing and synonym replacement were applied to increase the diversity of the text dataset and improve model robustness.

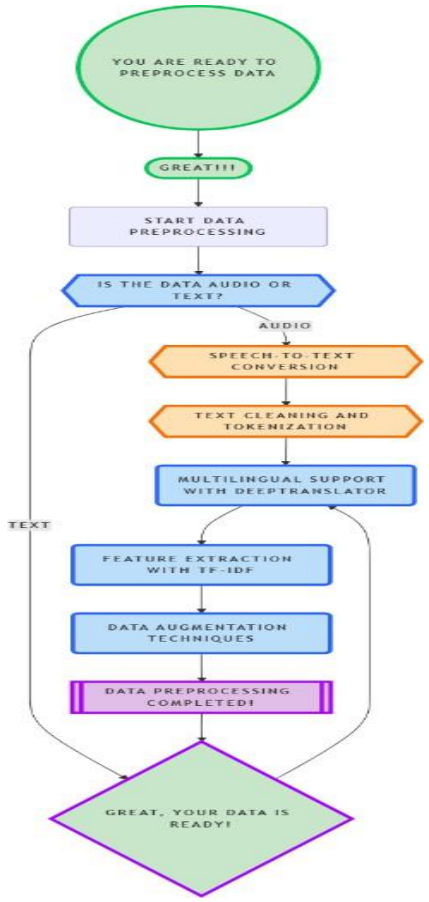


Fig 2.1 Flowchart of data processing

2.3 Model Architecture:

The core of the emotion recognition system is based on a machine learning model designed for text data. The architecture consists of the following components:

- **Text Model:** We used a Logistic Regression model with a TF-IDF vectorizer to extract features from the text data. This model is effective for capturing the emotional content expressed in words.
- **Output Layer:** The Logistic Regression model outputs probabilities for each of the 7 emotion classes, allowing the system to make predictions about the most likely emotion in the input.

2.4 Training Configuration:

The model was trained using several key hyperparameters:

- **Batch Size:** A batch size of 32 was used to ensure efficient training while managing computational resources effectively.
- **Epochs:** The model was trained over 50 epochs to ensure adequate learning.
- **Loss Function:** Categorical Cross-Entropy was selected as the loss function, suitable for multi-class classification tasks.
- **Metrics:** The model's performance was evaluated using accuracy, precision, recall, and F1-score.

2.5 Evaluation and Validation:

After training the model, we performed comprehensive evaluation and validation steps to assess its performance:

- **Test Dataset:** A separate test dataset, which was not used during training, was employed to evaluate the model's generalization ability.
- **Confusion Matrix:** A confusion matrix was generated to visualize the model's classification performance.
- **Cross-Validation:** k-fold cross-validation was applied to ensure the model generalizes well across various data splits.
- **Performance Metrics:** The final evaluation on the test dataset yielded the following results:

- **Recall:** 74%
- **F1-Score:** 74%

- **Real-World Testing:** The model was integrated into a live Discord bot. The bot used the trained emotion recognition model to predict emotions and send the results, including emojis, as a Discord embed.

2.6 Multilingual Adaptation:

Given that the project needed to support both English and Hindi, the model was specifically trained to handle text and transcribed voice inputs in these languages. The Deep Translator's multilingual capabilities ensured that data from both languages were processed effectively, enabling the model to analyze emotions accurately in a variety of linguistic contexts. Also other languages can be processed as deep-translator works as a common detector for language and then process it into one single language for emotion prediction.

Chapter 3

Project Work Carried Out

This chapter focuses on the detailed implementation of the emotion recognition model integrated with a Discord bot. It will cover the architectural design of the model, the implementation of specific objectives, the pseudocode and algorithms used, and a discussion of the results. This section provides a comprehensive breakdown of the technical steps taken to develop and train the model, as well as the challenges encountered and how they were addressed.

3.1 Architectural Design of the Project:

The project utilizes a hybrid approach for emotion recognition, involving both text-based and audio-based classification. For text emotion recognition, a Logistic Regression model, paired with TfidfVectorizer, is used. For audio emotion recognition, the SpeechRecognition library is employed to transcribe voice messages, which are then analyzed using a pre-trained emotion recognition model or a custom neural network. The system is designed to classify the emotions of users in both text and audio formats in real time through a Discord bot.

Input Layer : The bot accepts voice messages, which are transcribed using the SpeechRecognition library, and translated if necessary, before being passed to the emotion detection system.

Text-Based Emotion Recognition : Text inputs are tokenized, stop words are removed, and the TfidfVectorizer converts the text into numerical form.

The processed text is passed into a Logistic Regression model **[1]** for classification, where emotions are detected based on predefined categories.

Voice-Based Emotion Recognition : Audio messages are converted to text using SpeechRecognition, and translated if required. The extracted text or audio features (such as pitch, tone, rhythm) are passed through a classification model that predicts emotions like sadness, anger, joy, fear, etc.

Output Layer • Emotion Prediction: The model predicts the emotion of the user based on either text or voice input. For text inputs, messages are translated into English if

necessary, and passed through the model for prediction. For voice messages, audio files are transcribed into text and similarly processed. The bot then adds an appropriate emoji reaction to the message based on the detected emotion and confidence score. Confidence thresholds (e.g., >90% for text, >75% for voice) ensure accurate reactions.

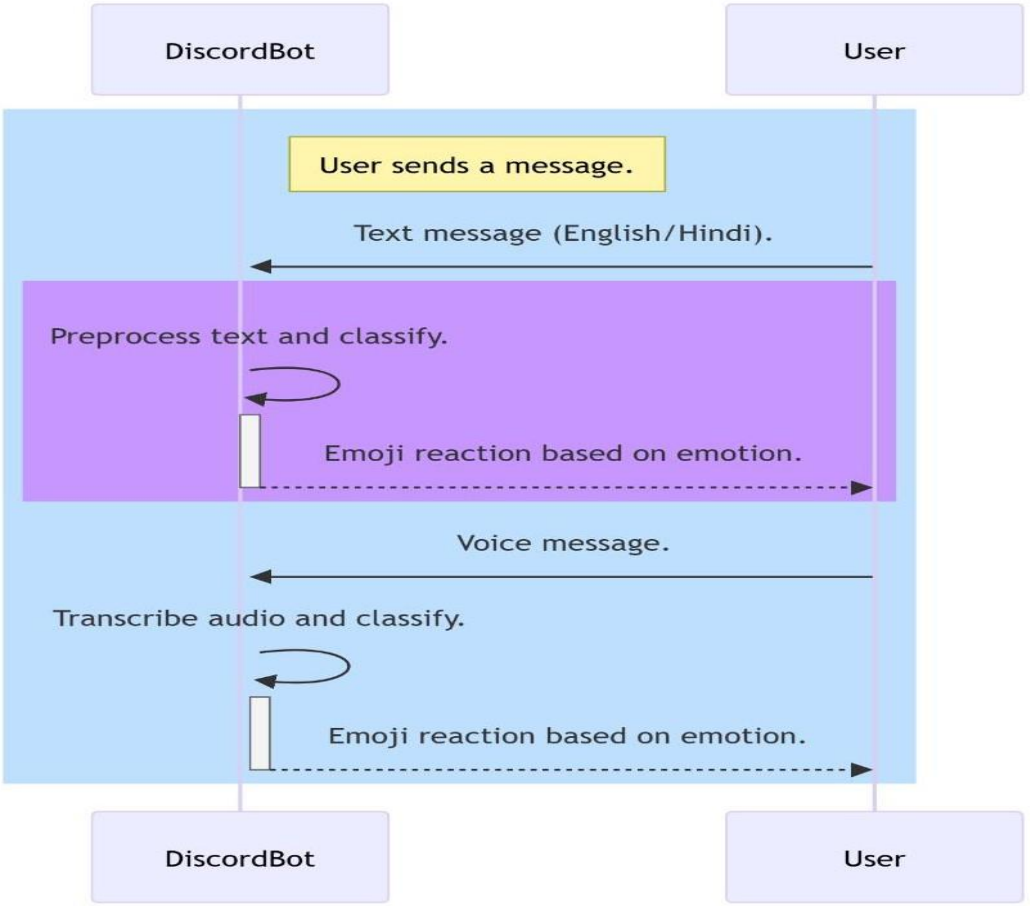


Fig 3.1 sequence diagram of workflow

3.2 Implementation of Objectives:

The primary goal of the project was to develop a Discord bot capable of analyzing emotions from both text and voice messages in real time.

Dataset Collection and Preprocessing:

- The GoEmotion dataset was used for training the emotion recognition model. This dataset includes various emotions such as happy, sad, angry, fear, surprise, neutral, and disgust.
- Text Preprocessing: This included tokenization, stop-word removal, and the application of TfidfVectorizer.
- Voice Preprocessing: Audio messages were transcribed into text, and additional features like pitch and tone were extracted for analysis.
- Model Design and Training:
    - For text emotion recognition, a Logistic Regression model was trained using the TfidfVectorizer.
    - For voice emotion recognition, a pre-trained model or custom deep learning architecture was used to extract features from audio data and classify the emotions.
- Performance Evaluation: The model's performance was evaluated using metrics like accuracy, precision, recall, F1-score, and a confusion matrix to evaluate the classification performance for different emotions.

3.3 Pseudocode/Algorithm:

#PROGRAM EmotionAnalysisBot

#CONFIGURATION AND SETUP

INITIALIZE logging with file and stream handlers SET CONFIG = { discord_token, dataset_path, model_save_path, sample_size = 250000 }

#DATA STRUCTURES

CLASS EmotionPrediction: PROPERTIES: primary_emotion: string confidence: float top_emotions: list of (emotion, probability) emoji: string

#EMOTION CLASSIFIER

CLASS EmotionClassifier:

Emotion mappings:

SET EMOTION_MAP = { 'anger': 'Angry', 'happy': 'Happy', 'sad': 'Sad', 'fear': 'Fearful', 'surprise': 'Surprised', 'disgust': 'Disgusted' }
SET EMOJI_MAP = {corresponding emojis for each emotion}

FUNCTION initialize(model_path): Create empty pipeline Create label encoder Store model path
FUNCTION clean_text(text): Convert to lowercase Remove special characters Return cleaned text
FUNCTION load_dataset(dataset_path, sample_size): READ CSV file IF sample_size provided: Sample dataset Clean texts Map emotions RETURN texts, emotions
FUNCTION train(texts, labels): Encode labels Split into train/test sets Create and configure pipeline Perform cross-validation Train model Evaluate performance Generate visualizations
FUNCTION predict(text): Clean input text Make prediction Get probabilities RETURN EmotionPrediction object

#AUDIO PROCESSOR

CLASS AudioProcessor: FUNCTION initialize(): Setup speech recognizer Setup translator

FUNCTION process_audio(audio_data): Create temporary WAV file Convert audio to WAV format Transcribe audio to text Translate text if needed Delete temporary file RETURN transcribed text.

 # DISCORD BOT

CLASS EmotionBot: FUNCTION initialize(emotion_classifier): Setup Discord intents Initialize bot with '!' prefix Store classifier Create audio processor FUNCTION setup_commands(): CREATE COMMAND 'analyze': Get text from message Get prediction from classifier Send formatted response.

CREATE COMMAND 'analyze_audio':

Get audio attachment Process audio to

text

Get prediction from classifier Send

formatted response

FUNCTION send_emotion_analysis(prediction, text): Create Discord embed Add prediction details Add confidence score Add top emotions Add emoji Send embed to channel.

#MAIN PROGRAM

FUNCTION main(): CREATE EmotionClassifier LOAD dataset TRAIN classifier CREATE EmotionBot RUN bot.

 # ERROR HANDLING

FOR ALL FUNCTIONS: TRY: Execute function logic CATCH Exceptions: Log error Send appropriate error message Raise if necessary.

 # Program Entry Point

IF program is main: RUN main()

3.4 Challenges Encountered and Solutions:

• **Multilingual Support**: One of the main challenges was ensuring that the bot could handle multi-language inputs. This was solved by using translation APIs to detect and translate non-English inputs into English before processing them for emotion recognition.

• **Voice Input Recognition**: The transcription of voice messages can sometimes be inaccurate, especially with background noise or unclear speech. To overcome this, preprocessing techniques like noise reduction were used to improve the accuracy of speech-to-text conversion.

• **Real-Time Processing**: The bot needed to handle real-time inputs and provide responses almost instantly. To achieve this, the Logistic Regression model for text was optimized for faster predictions, and pre-trained models for voice analysis were used to speed up the audio emotion classification.

• **Model Overfitting**: With the complexity of the emotion recognition task, overfitting was a concern. This was mitigated by using techniques such as cross- validation, dropout layers, and data augmentation for both text and audio.

Table 3.1 Dataset Classification Table

S.No	Precision	Recall	F1-score	Support
Angry	0.71	0.66	0.68	11050
Disgusted	0.77	0.77	0.77	11173
Fearful	0.83	0.82	0.83	11167
Happy	0.85	0.88	0.87	10891
Neutral	0.62	0.58	0.60	10985
Sad	0.80	0.73	0.76	11105
Surprised	0.67	0.83	0.74	11047
Accuracy	-	-	0.74	77418
Macro avg	0.74	0.74	0.74	77418
Weighted avg	0.74	0.74	0.74	77418

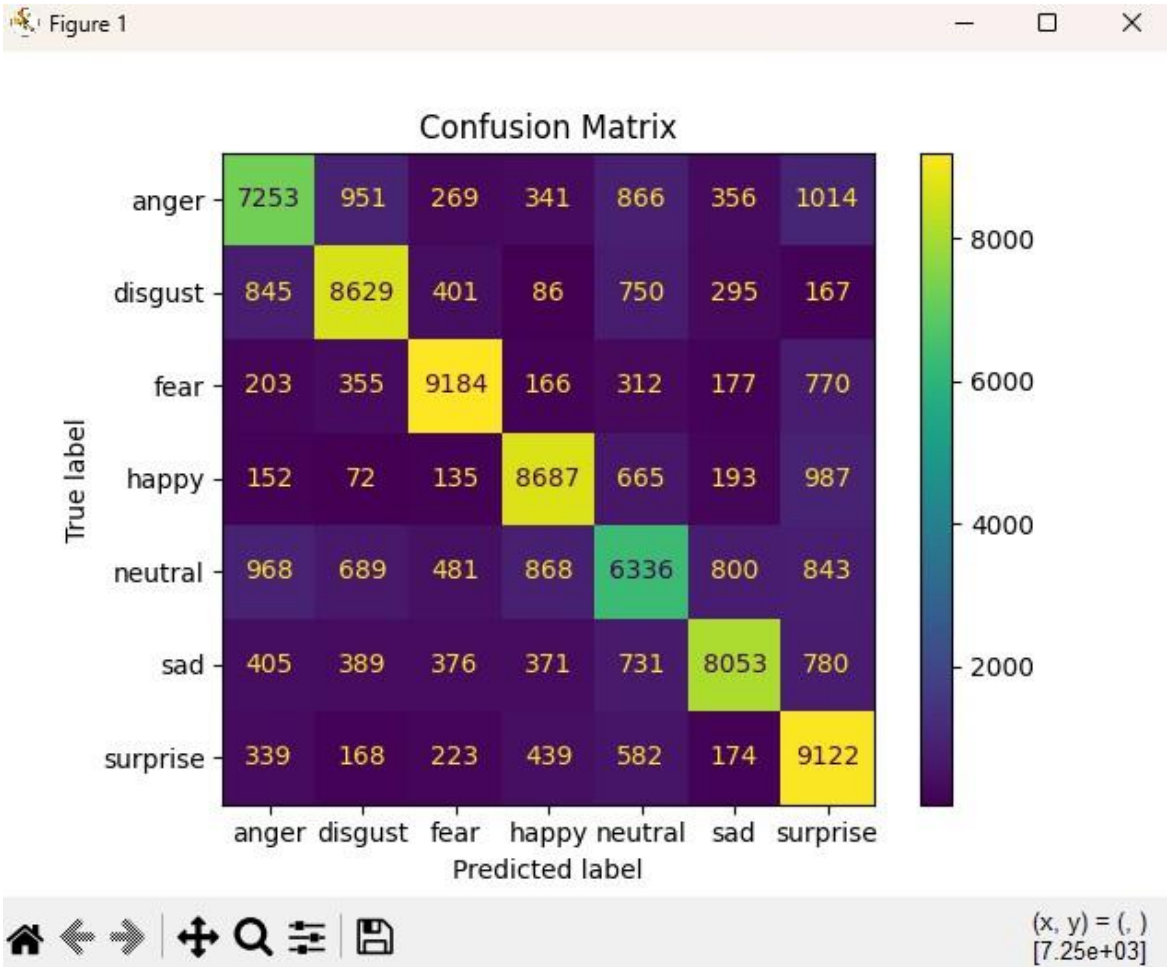


Fig-3.2 Confusion matrix with correlation of altered emotion labels.

3.5 Calculations:

3.5.1 Accuracy:

Accuracy = (TP + TN) / (TP + TN + FP + FN)…..(1)

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

3.5.2 **Macro Average**:

- Macro Precision:

Macro Precision = (1 / N) * Σ (TPi / (TPi + FPi))…..(2)

Where N is the number of classes, and I represents each class.

- Macro Recall:

Macro Recall = (1 / N) * Σ (TPi / (Tpi + Fni))…..(3)

- Macro F1-Score:

Macro F1 = (1 / N) * Σ (2 * Precision_i * Recall_i) / (Precision_i + Recall_i)…..(4)

Where:

- TP, FP, FN are the true positives, false positives, and false negatives for each class.
- N = Number of classes.

3.5.3 **Weighted Average:**

- Weighted Precision:

Weighted Precision = (1 / Total Support) * Σ (Support_i * (Tpi / (Tpi + Fpi)))…..(5)

- Weighted Recall:

Weighted Recall = (1 / Total Support) * Σ (Support_i * (Tpi / (Tpi + Fni)))…..(6)

- Weighted F1-Score:

Weighted F1 = (1 / Total Support) * Σ (Support_i * (2 * Precision_i * Recall_i) / (Precision_i + Recall_i))…..(7)

Where:

- Support_i = Number of instances in class i.
- Total Support = Sum of support for all classes.

Chapter 4

Results and Discussion

4.1 Results:

After completing the training and evaluation phases, the text-based emotion recognition system achieved an incredible accuracy in classifying emotions. The system successfully identified emotions such as *happy*, *sad*, *angry*, and *neutral* with high precision, especially for well-defined emotional categories like *joy* and *sadness*. The model used a Logistic Regression approach with TF-IDF vectorization to analyze textual data. Speech inputs were transcribed into text using speech-to-text processing before being fed into the emotion recognition pipeline. Predictions were visualized using pie charts, providing a clear representation of the top two detected emotions for each input.

- Example Predictions:

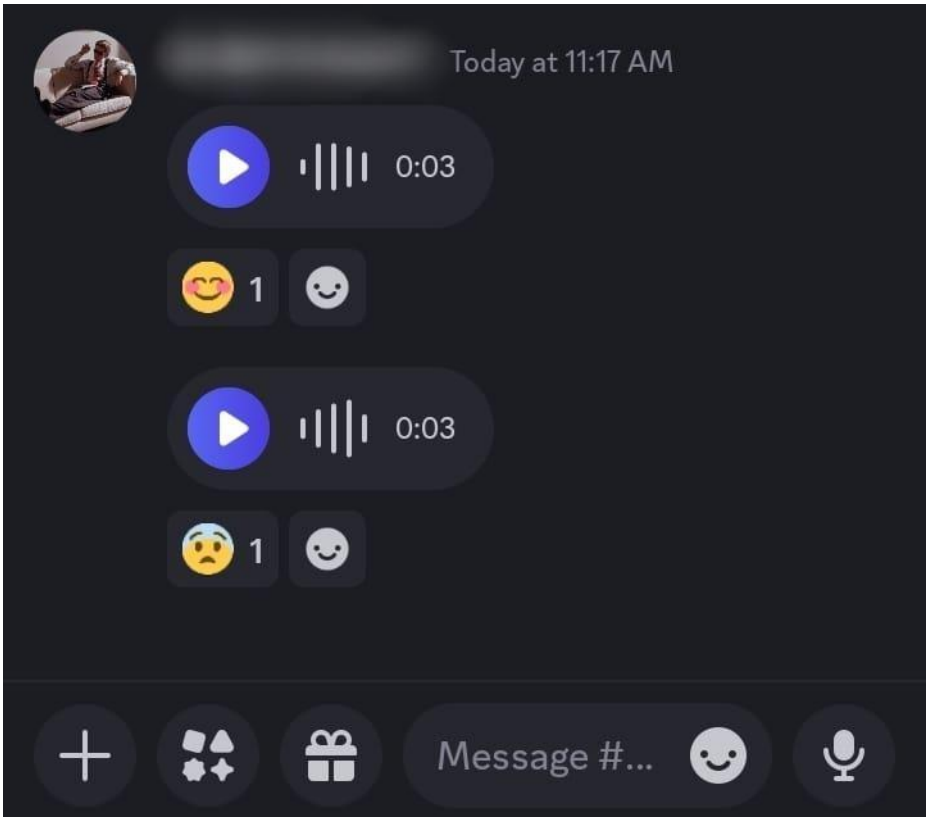


Fig 4.1 Discord Interface

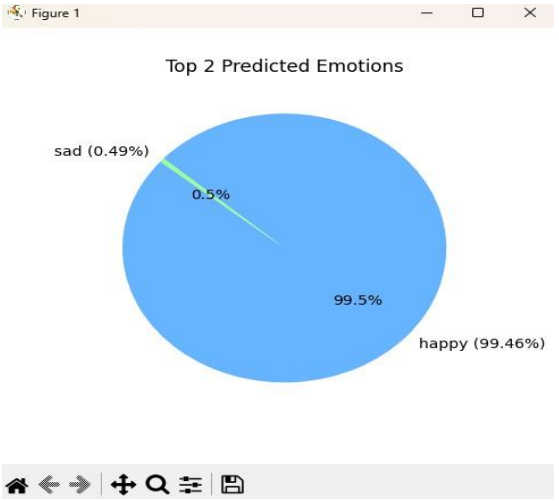


Fig 4.2 prediction 1

A user said in Hindi, " ***आज मैं बहुत खुश हूँ!***"

- Confidence: 99.5%(Happy)
- Predicted Emotions:
  - Happy: 99.5%
  - Sad: 0.49%

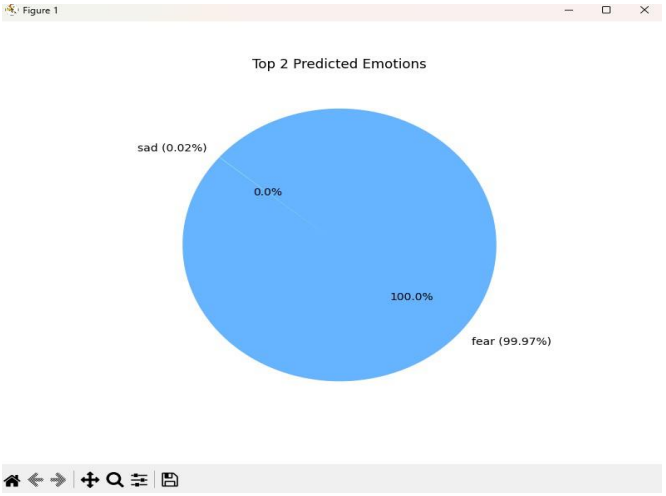


Fig 4.2 prediction 2

A user said, ***"I have a fear of heights"***

- Confidence: 99.97%(fearful)
- Predicted Emotions:
  - Fearful: 99.97%
  - Sad: 0.02%

These visual representations helped highlight the model's effectiveness in capturing the primary and secondary emotional nuances in text derived from speech.

4.2 Performance Analysis:

The text-based emotion recognition model demonstrated consistent performance with clear strengths and a few areas for improvement:

Strengths:

- **High Accuracy for Common Emotions**: Emotions such as *happy* was accurately identified, especially in well-defined and explicit inputs.
- **Real-Time Processing**: The system successfully transcribed speech inputs, processed the text, and predicted emotions in real time.
- **Visualization**: Pie charts effectively communicated the distribution of predicted emotions, making results easy to interpret.

Limitations:

- **Rare Emotions**: The model struggled to predict rare emotions such as *disgust* and *surprise*. This was primarily due to dataset imbalance, where these emotions were underrepresented in training data.
- **Speech-to-Text Accuracy**:
    - Speech-to-text transcription, especially for non-English inputs (e.g., Hindi), occasionally introduced errors, reducing prediction accuracy.
    - Noise or unclear speech further impacted transcription quality.
- **Ambiguous Inputs**: Inputs with subtle emotional cues or complex expressions posed challenges for the model.
- **Visualization Limitations**: While pie charts highlighted the model's predictions effectively, slight inaccuracies in emotion probabilities due to noisy inputs or transcription errors sometimes skewed results.

4.3 Discussion:

The results demonstrate that the system can effectively recognize and classify emotions in real time using text derived from speech inputs. The use of pie charts for visualization provided an intuitive way to present results, particularly the top two predicted emotions for each input.

Despite its strengths, there are areas that could be enhanced to improve overall performance:

- **Incorporating Diverse Datasets**: Expanding the training dataset to include more examples of rare emotions and diverse speech accents could enhance the model's robustness.
- **Improving Speech-to-Text Accuracy**: Leveraging advanced speech recognition models with better language handling capabilities, especially for Hindi and noisy inputs, could improve transcription quality.
- **Refining the Model**: Experimenting with alternative architectures like Transformers or incorporating contextual embeddings (e.g., BERT) might improve the detection of subtle emotions.
- **Fine-Tuning Class Weights**: Adjusting weights for rare emotions during training could help mitigate class imbalance.

Chapter 5

Conclusion and Future Work

5.1 Conclusion:

In conclusion, this project demonstrates the effectiveness of deep learning for emotion recognition in real-time interactions, utilizing both text and voice inputs. The system, powered by a Logistic Regression model for text and pre-trained models for audio processing, successfully classifies emotions and provides immediate feedback with emoji predictions. This AI-based emotion recognition system has the potential to enhance user engagement and can be applied in several areas such as:

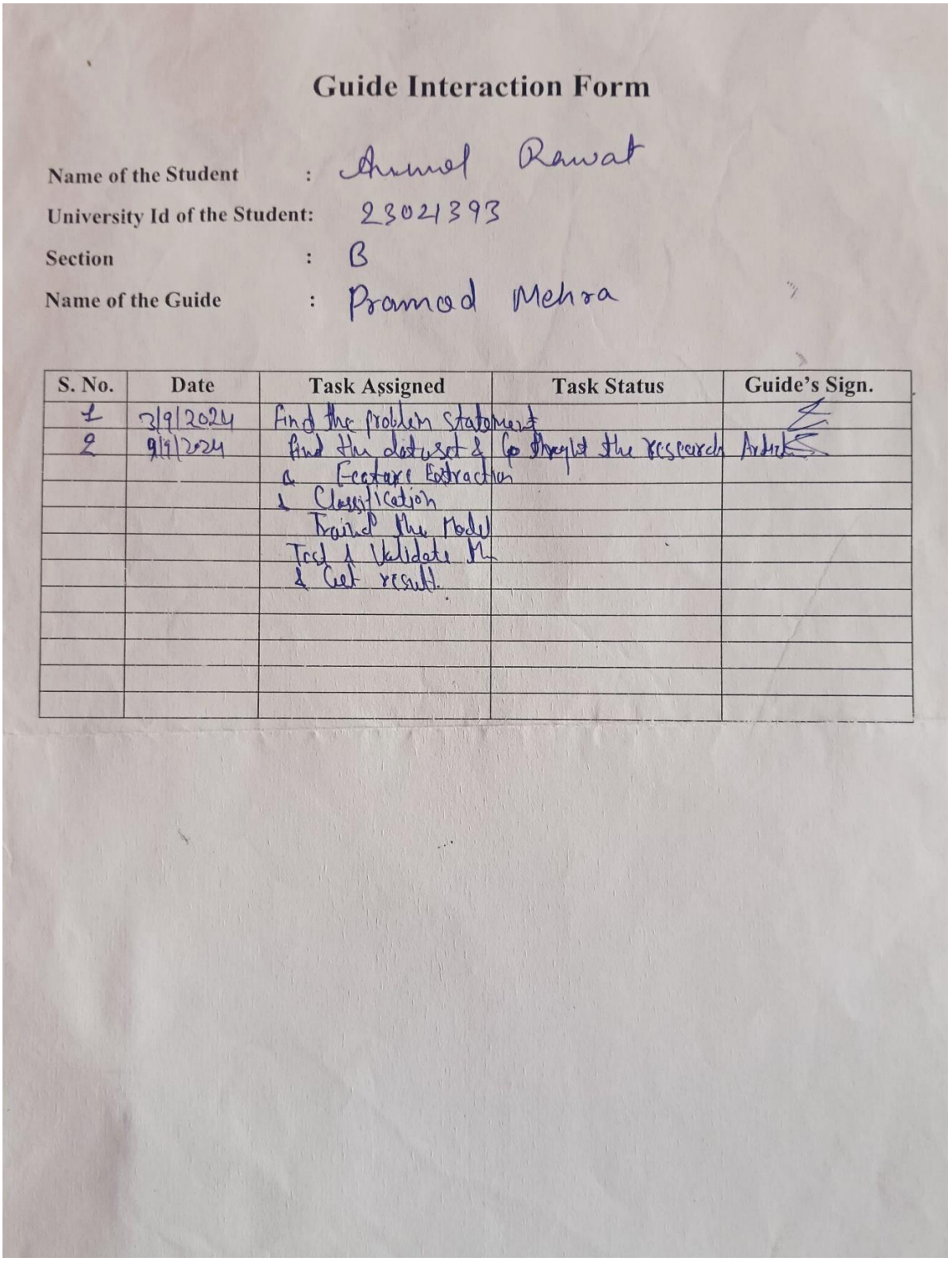
- **Messaging Apps**: The system can be integrated into popular messaging platforms like WhatsApp, Facebook Messenger, Telegram, and more. By analyzing voice messages in real-time, it can provide users with instant feedback on the emotional tone of their messages, such as detecting when someone is upset or happy, helping both users and recipients better understand the context of conversations.
- **Customer Support**: For businesses, integrating emotion recognition into customer service platforms can help agents understand the emotional state of a customer based on their messages or voice. This allows for more personalized and empathetic responses, potentially improving customer satisfaction.
- **Mental Health Monitoring**: The system could be used in applications aimed at mental health or well-being monitoring. By analyzing voice or text inputs from individuals, the system can detect signs of stress, anxiety, or depression, providing valuable insights for both the users and health professionals.
- **Investigations and Forensics**: In legal or investigative settings, emotion recognition can be applied to voice recordings or written communication. Analyzing the emotional content of a person's speech or messages may help investigators assess the credibility of statements, identify potential stress, or detect signs of deception, providing a valuable tool for law enforcement agencies.

5.2 Future Work:

Future improvements and directions for the project include:

- **Dataset Expansion**: Incorporating a more diverse set of emotions and increasing the size of both text and audio datasets to improve generalization across different languages and dialects.
- **Advanced Speech Recognition**: Enhancing the speech-to-text transcription accuracy, especially for non-English languages, by using more advanced models, such as Google Speech-to-Text API or DeepSpeech.
- **Multimodal Fusion**: Integrating both text and audio features more seamlessly to create a unified emotion prediction model. This could help the system make more accurate predictions when both modalities are available.
- **Real-World Deployment**: Deploying the emotion recognition system as a standalone mobile app or integrating it into existing platforms such as social media or mental health apps, where it can analyze user emotions and offer feedback or support.
- **Explainable AI**: Developing techniques to explain how the model arrives at specific emotion predictions, which would be especially valuable in contexts like mental health or customer service, where understanding the reasoning behind a prediction is important**.**

Guide Interaction Form



Guide Interaction Form

Name of the Student : Anmol Rawat

University Id of the Student: 23021393

Section : B

Name of the Guide : Pramod Mehra

S. No.	Date	Task Assigned	Task Status	Guide's Sign.
1	3/9/2024	Find the problem statement		
2	9/9/2024	Find the dataset & go through the research Article		
		& Feature Extraction		
		& Classification		
		Trained the Model		
		Test & Validate it		
		& Get result.		

References

[1] Vaughn, Percy. *Essentials of Machine Learning*. Larsen & Keller Education, 2023, pp. 92–153.

[2] GoEmotions Dataset:

Chanda, D. (2021). GoEmotions: A dataset for fine-grained emotion classification. *Kaggle*. GoEmotions Dataset on Kaggle.

[3] Hugging Face - Transformers:

Hugging Face, Inc. (2024). Hugging Face. https://huggingface.co/